

An LP with Integrality Gap $1 + \epsilon$ for Multidimensional Knapsack

David Pritchard*

February 3, 2011

Abstract

In this note we study packing or covering integer programs with at most k constraints, which are also known as *k-dimensional knapsack problems*. For integer $k > 0$ and real $\epsilon > 0$, we observe there is a polynomial-sized LP for the k -dimensional knapsack problem with integrality gap at most $1 + \epsilon$. The variables may be unbounded or have arbitrary upper bounds. In the (classical) packing case, we can also remove the dependence of the LP on the cost-function, yielding a polyhedral approximation of the integer hull. This generalizes a recent result of Bienstock [3] on the classical knapsack problem.

1 Introduction

The classical *knapsack problem* is the following: given a collection of items each with a value and a weight, and given a weight limit, find a subset of items whose total weight is at most the weight limit, and whose value is maximized. If n denotes the number of items, this can be formulated as the integer program $\{\max \sum_{i=1}^n x_i v_i \mid x \in \{0, 1\}^n, \sum_{i=1}^n x_i w_i \leq \ell\}$ where n denotes the number of items, v_i denotes the value of item i , w_i denotes the weight of item i , and ℓ denotes the weight limit.

In the more general *k-dimensional knapsack* (or *k-constrained knapsack*) problem, there are k different kinds of “weight” and a limit for each kind. An example for $k = 3$ would be a robber who is separately constrained by the total mass, volume, and noisiness of the items he is choosing to steal. An orthogonal generalization is that the robber could take multiple copies of each item i , up to some prescribed limit of d_i available copies. We therefore model the k -dimensional knapsack problem as

$$\{\max cx \mid x \in \mathbb{Z}^n, 0 \leq x \leq d, Ax \leq b\} \quad (1)$$

where A is a k -by- n matrix, b is a vector of length k , and d is a vector of length n , all non-negative and integral. Two special cases are common: if $d = \mathbf{1}$ we call it the *0-1 knapsack problem*; if $d = +\infty$, we call it the *unbounded knapsack problem*.

Another natural generalization is the *k-dimensional knapsack-cover problem*,

$$\{\min cx \mid x \in \mathbb{Z}^n, 0 \leq x \leq d, Ax \geq b\}$$

which has analogous unbounded and 0-1 special cases. We sometimes call this version the *covering version* and likewise (1) is the *packing version*.

*École Polytechnique Fédérale de Lausanne and partially supported by an NSERC post-doctoral fellowship.

On the positive side, for any fixed k , all above variants admit a simple pseudo-polynomial-time dynamic programming solution. Chandra et al. [9] gave the first PTAS (polynomial-time approximation scheme) for k -dimensional knapsack in 1976, and later an LP-based scheme was given by Frieze and Clarke [13]. See the book by Kellerer et al. [18, §9.4.2] for a more comprehensive literature review. The case $k = 1$ also admits a fully polynomial-time approximation scheme (FPTAS), but for $k \geq 2$ there is no FPTAS unless $P=NP$. This was originally shown for 0-1 k -dimensional knapsack by Gens & Levner [15] and Korte & Schrader [20] (see also [18]) and subsequently for arbitrary d by Magazine & Chern [24].

Our main result is the following:

Theorem 1. *Let k and ϵ be fixed. Given a k -dimensional knapsack (resp. knapsack-cover) instance \mathcal{K} , there is a polynomial-sized extended LP relaxation \mathcal{L} of \mathcal{P} with $\text{OPT}(\mathcal{P}) \geq (1 - \epsilon)\text{OPT}(\mathcal{L})$ (resp. with $\text{OPT}(\mathcal{P}) \leq (1 + \epsilon)\text{OPT}(\mathcal{L})$).*

Here “polynomial-sized extended LP relaxation” means the following. First, \mathcal{P} has n variables. Then \mathcal{L} must have those n variables plus a polynomial number of other ones. The projection \mathcal{L}' of \mathcal{L} onto the first n variables must contain the same integral solutions as \mathcal{P} . Finally, \mathcal{L} and \mathcal{P} must have the same objective function, i.e. the objective function should ignore the extended variables.

In the proof, we will see that the LP can be constructed in polynomial time, and that a near-optimal integral solution can be obtained from an optimal extreme point fractional solution just by rounding down (resp. up). The number of variables in the LP is $n^{O(k/\epsilon)}$ and the number of constraints is $kn^{O(k/\epsilon)}$. The *integrality gap* of an IP is the worst-case ratio between the fractional and integral optimum and therefore Theorem 1 can be equivalently stated as saying that \mathcal{P} has integrality gap at most $1 + \epsilon$.

Our result and the techniques we use are a generalization of a recent result of Bienstock [3], which dealt with the packing version for $k = 1$. The key observation we contribute is that his “filtering” approach was also traditionally used to get a PTAS for multi-dimensional knapsack; in *filtering* we exhaustively guess the γ max-cost items in the knapsack for some constant γ .

The construction of \mathcal{L} in Theorem 1 turns out to depend on the cost function c . A more interesting and challenging problem is to find an \mathcal{L} which is independent of the cost-function, since this gives a *polyhedral approximation* \mathcal{L}' of \mathcal{P} e.g. in the packing case, it implies $\mathcal{L}' \supset \mathcal{P} \supset (1 - \epsilon)\mathcal{L}'$. Bienstock’s result [3] actually gives an LP which does not depend on the item cost/profits c . We will show (in Section 4) that in the packing case, our approach can be similarly revised:

Theorem 2. *Let k and ϵ be fixed. Given a k -dimensional knapsack instance \mathcal{K} , there is a polynomial-sized extended LP relaxation \mathcal{L} of \mathcal{P} with $\text{OPT}(\mathcal{P}) \geq (1 - \epsilon)\text{OPT}(\mathcal{L})$, such that \mathcal{L} does not depend on c .*

This comes at the cost of an increase in size to $kn^{O(k^2/\epsilon)}$. For the covering case performing the same (a polynomial-sized extended LP relaxation independent of c with integrality gap $\leq 1 + \epsilon$) is an interesting open problem; we elaborate at the end.

1.1 Related Work

Knapsack (whether packing or covering) has an FPTAS by dynamic programming, and it is well-known that dynamic programs of such a form can be solved as a shortest-path problem, which has an LP formulation. Nonetheless, there is no evident way to combine these steps to get an LP for

knapsack with integrality gap $1 + \epsilon$. The problem (say, for packing, which is simpler) is that last step in the FPTAS is not merely to return the last entry of the DP table, but rather it finds the maximum scaled profit such that the minimum volume to obtain it fits inside the knapsack (and then recovers the actual solution). The naive fix is adding this volume constraint to the LP but it makes the LP non-integral and then it is not clear how to proceed.

Bienstock & McClosky [4] extend the work of Bienstock [3] to covering problems and other settings, and also give an LP of size $n^2(1/\epsilon)^{\frac{1}{\epsilon} \log \frac{1}{\epsilon}}$ with integrality gap $1 + \epsilon$ for 1-dimensional, 0-1 covering knapsack.¹ There is some current work [5] on obtaining primal-dual algorithms (that is, not needing the ellipsoid method or interior-point subroutines) for knapsack-type covering problems with good approximation ratio and [4] reports that the methods of [5] extend to a combinatorial LP-based approximation scheme for 1-dimensional covering knapsack.

Answering an open question of Bienstock [4] about the efficacy of automatic relaxations for the knapsack problem, Karlin et al. [17] recently found that the “Lasserre hierarchy” of semidefinite programming relaxations, when applied to the 1-dimensional 0-1 packing knapsack problem, gives an SDP with integrality gap $1 + \epsilon$ after $O(1/\epsilon^2)$ rounds.

Knapsack problems have a couple of interesting basic properties. The first contrasts with Lenstra’s result [22] that for any fixed k , integer programs with k constraints can be solved in polynomial time; in comparison, if we have nonnegativity constraints for every variable plus *one other constraint*, we get the unbounded (1-dimensional) knapsack problem, which is NP-hard [23]. Second, recall that for any optimization problem whose objective is integral, and whose optimal value is polynomial in the input size, any FPTAS can be used to get a pseudopolynomial-time algorithm. In contrast, 0-1 2-dimensional knapsack shows the converse is false: it has a pseudopolynomial-time algorithm, but getting an FPTAS is NP-hard even when each profit c_i is 1, e.g. see [18, Thm. 9.4.1].

There is a line of work on maximizing constrained submodular functions. For non-monotone submodular maximization subject to k linear packing constraints, the state of the art is by Lee et al. [21] who give a $(5 + \epsilon)$ -approximation algorithm. For monotone submodular maximization the state of the art is by Chekuri & Vondrák [10] who give a $(e/(e - 1) + \epsilon)$ -approximation subject to k knapsack constraints and a matroid constraint. We note it is NP-hard to obtain any factor better than $e/(e - 1)$ for monotone submodular maximization over a matroid [12], so in this setting knapsack constraints only affect the best ratio by ϵ , just like in our setting of LP-relative approximation.

1.2 Overview

First, we review rounding and filtering. Rounding is a standard approach to turn an optimal fractional solution into a nearly-optimal integral one, and here we lose up to k times the maximum per-item profit. Filtering works well with rounding because it reduces the maximum per-item profit; the power of these ideas is already enough to get an LP-based approximation scheme [13], but it uses a separate LP for each “guess” made in filtering. Therefore, like Bienstock [4], we use disjunctive programming [2] to combine all the separate LPs into a single one. The approach has some similarity to the knapsack-cover inequalities of Carr et al. [6].

¹They use a disjunctive program; in essence, the LP guesses the most costly item in the knapsack, then for $i = 1, \dots, O(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$ it guesses the number of items whose costs are $(1 + \frac{1}{\epsilon})^{-(i, i+1]}$ times that cost, with all guesses $> \frac{1}{\epsilon}$ deemed equivalent. In particular the LP depends on the cost function. We remark that the method does not readily extend to k -dimensional knapsack.

2 Rounding and Filtering

We now explain the approach. A knapsack instance (1) is determined by the parameters (A, b, c, d) . The naïve LP relaxation of the knapsack problem is

$$\{\max cx \mid x \in \mathbb{R}^n, 0 \leq x \leq d, Ax \leq b\}. \quad \mathcal{K}(A, b, c, d)$$

In the following, *fractional* means non-integral. The following lemma is standard.

Lemma 3. *Let x^* be an extreme point solution to the linear program $(\mathcal{K}(A, b, c, d))$. Then x^* is fractional in at most k coordinates.*

Proof. It follows from elementary LP theory that $x^* \in \mathbb{R}^n$ satisfies n (linearly independent) constraints with equality. There are k constraints of the form $A_j x \leq b_j$; all other constraints are of the form $x_i \geq 0$ or $x_i \leq d_i$, so at least $n - k$ of them hold with equality. Clearly $x_i \geq 0$ and $x_i \leq d_i$ cannot both hold with equality for the same i , so $x_i^* \in \{0, d_i\}$ for at least $n - k$ distinct i , which gives the result. \square

Therefore, we obtain the following primitive guarantee on a rounding strategy. Let $\lfloor \cdot \rfloor$ applied to a vector mean component-wise floor and let $c_{\max} := \max_i c_i$.

Corollary 4. *Let x^* be an extreme point solution to the linear program $(\mathcal{K}(A, b, c, d))$. Then $c \lfloor x^* \rfloor \geq cx^* - kc_{\max}$.*

Now the idea is to take x^* to be an optimal fractional solution, and use filtering (exhaustive guessing) to turn the additive guarantee into a multiplicative factor of $1 + \epsilon$. Let γ denote a parameter, which represents the size of a multi-set we will guess. For a non-negative vector z let the notation $\|z\|_1$ mean $\sum_i z_i$. A *guess* is an integral vector g with $0 \leq g \leq d, Ag \leq b$ and $\|g\|_1 \leq \gamma$. It is easy to see the number of possible guesses is bounded by $(n + 1)^\gamma$, and that for any constant γ we can iterate through all guesses in polynomial time.

From now on we assume without loss of generality (by reordering items if necessary) that $c_1 \leq c_2 \leq \dots \leq c_n$. For a guess g with $\|g\|_1 = \gamma$ we now define the *residual knapsack problem* for g . The residual problem models how to optimally select the remaining objects *under the restriction* that the γ most profitable² items chosen (counting multiplicity) are g . Let $\mu(g)$ denote $\min\{i \mid g_i > 0\}$. Define d^g to be the first $\mu(g)$ coordinates of $d - g$ followed by $n - \mu(g)$ zeroes, and $b^g = b - Ag$. The *residual knapsack problem* for g is (A, b^g, c, d^g) . The residual problem for g does not permit taking items with index more than $\mu(g)$ and so its c_{\max} value may be thought of as $c_{\mu(g)}$ or less, which is at most $c \cdot g / \|g\|_1 = c \cdot g / \gamma$.

If a guess g has $\|g\|_1 < \gamma$, define b^g and d^g to be all-zero. Then Corollary 4 gives the following.

Corollary 5. *Let x_{OPT} be an optimal integral knapsack solution for (A, b, c, d) . Let g be the γ most profitable items in x_{OPT} (or all, if there are less than γ). Let x^* be an optimal extreme point solution to $\mathcal{K}(A, b^g, c, d^g)$. Then $g + \lfloor x^* \rfloor$ is a feasible knapsack solution for (A, b, c, d) with value at least $1 - k/\gamma$ times optimal.*

²To simplify the description, even if $c_{i+1} = c_i$ we think of item $i + 1$ as more profitable than item i .

Proof. We use OPT to denote $c \cdot x_{\text{OPT}}$. Note that $x_{\text{OPT}} - g$ is feasible for the residual problem for g . Therefore $c \cdot x^* \geq \text{OPT} - c \cdot g$. Moreover c_{\max} in the residual problem for g is not more than $c \cdot g / \gamma \leq \frac{\text{OPT}}{\gamma}$, so Corollary 4 shows that

$$c \cdot \lfloor x^* \rfloor \geq c \cdot x^* - k \frac{\text{OPT}}{\gamma} \geq \text{OPT} - c \cdot g - k \frac{\text{OPT}}{\gamma}$$

and consequently $\lfloor x^* \rfloor + g$ is a solution with value at least $\text{OPT}(1 - \frac{k}{\gamma})$, as needed. \square

By taking $\gamma = k/\epsilon$ and solving $\mathcal{K}(A, b^g, c, d^g)$ for all possible g we get the previously known PTAS for k -dimensional knapsack; we now refine the approach to get a single LP.

3 Disjunctive Programming

We now review some disjunctive programming tools [2]. The only result we need is that it is possible to write a compact LP for the convex hull of the union of several polytopes, provided that we have compact LPs for each one.

Suppose we have polyhedra $P^1 = \{x \in \mathbb{R}^n \mid A^1 x \leq b^1\}$ and $P^2 = \{x \in \mathbb{R}^n \mid A^2 x \leq b^2\}$. Both of these sets are convex and it is therefore easy to see that the convex hull of their union is the set

$$\text{conv.hull}(P^1 \cup P^2) = \{x \in \mathbb{R}^n \mid x = \lambda x^1 + (1 - \lambda)x^2, 0 \leq \lambda \leq 1, A^1 x^1 \leq b^1, A^1 x^2 \leq b^2\}.$$

However, this is not a *linear* program, e.g. since we multiply the variable λ by the variables x^1 . Nonetheless, it is not hard to see that the following is a linear formulation of the same set:

$$\text{conv.hull}(P^1 \cup P^2) = \{x \in \mathbb{R}^n \mid x = x^1 + x^2, 0 \leq \lambda \leq 1, A^1 x^1 \leq \lambda b^1, A^1 x^2 \leq (1 - \lambda)b^2\}.$$

A similar construction gives the convex hull of the union of any number of polyhedra; we now apply this to the knapsack setting.

The LP $\mathcal{K}(A, b^g, c, d^g)$ was constructed to mean the left-over problem after making a guess g of the γ most profitable items; we similarly shift this LP to get $\{y = x + g \mid x \in \mathbb{R}^n, 0 \leq x \leq d^g, Ax \leq b^g\}$ which is the same set, after the guessed part is added back in.

Let \mathcal{G} denote the set of all possible guesses g . Then the convex hull of the union of the shifted polyhedra is given by the feasible region of the following polyhedron:

$$\left\{ y \mid y = \sum_{g \in \mathcal{G}} y^g; \sum_{g \in \mathcal{G}} \lambda^g = 1; \lambda \geq \mathbf{0}; \forall g : y^g = x^g + \lambda^g g, \mathbf{0} \leq x^g \leq \lambda^g d^g, A y^g \leq \lambda^g b^g \right\}. \quad (\mathcal{L})$$

We attach objective $\max c \cdot y$ to (\mathcal{L}) to make it into an LP, and use it to prove Theorem 1.

Proof of Theorem 1, packing version. Let y be an optimal extreme point solution for (\mathcal{L}) . It is straightforward to argue that any extreme point solution has $\lambda^{g^*} = 1$ for some particular g^* , and $\lambda^g = 0$ for all other g . Hence $y = x^{g^*} + g^*$ where x^{g^*} is an optimal extreme point solution to $\mathcal{K}(A, b^{g^*}, c, d^{g^*})$. We now show that $\lfloor y \rfloor$ is a $(1 - \epsilon)$ -approximately optimal solution, re-using the previous arguments.

If $\|g^*\|_1 < \gamma$, then $x^{g^*} = 0$ so y is integral, hence y is an optimal knapsack solution. Otherwise, if $\|g^*\|_1 = \gamma$, then Corollary 4 shows that

$$c \cdot \lfloor y \rfloor = c \cdot \lfloor x^{g^*} \rfloor + c \cdot g^* \geq c \cdot x^{g^*} - k \frac{c \cdot g^*}{\gamma} + c \cdot g^* = c \cdot y - k \frac{c \cdot g^*}{\gamma} \geq (1 - \epsilon) c \cdot y,$$

which completes the proof. \square

The corresponding result for the covering version is very similar. One difference is that we round up instead of down. The other is that some guesses become inadmissible. Let g be an integral vector with $0 \leq g \leq d, \|g\|_1 \leq \gamma$; we define $\mu(g), d^g$ as before and call g a *guess* only if $A(g + d^g) \geq b$, in which case we set b^g to be the component-wise maximum of $\mathbf{0}$ and $b - Ag$.

4 Removing the Dependence on c for Packing Problems

In the LPs described above, for each guess g , we treated that guess as the set of most *profitable* items. In particular, b^g and d^g are defined in a way that depends on c . We now show in the packing case, how to write a somewhat larger LP, still with integrality gap $1 + \epsilon$, which is defined independently of c . This exactly follows the approach of Bienstock [3]; what we will do is guess the *biggest* items for each constraint, rather than the most profitable items. The technique does not seem to have an easy analogue for covering problems.

In detail, previously, we guessed the multiset g of γ most profitable items in the solution. Instead, let us guess a k -tuple (g^1, g^2, \dots, g^k) where for each k , g^i is the set of γ items in the solution which have largest coefficients with respect to the i th constraint (breaking ties in each constraint in any consistent way). What we need is that any extreme feasible solution with at most k fractional values can be rounded to an integral feasible solution at a relative cost factor of at most ϵ . Let the original extreme point LP solution be x . We round each fractional value up to the closest integer, which causes the solution to become an infeasible one, call it $y = \lceil x \rceil$. Then, to retain feasibility, we go through each of the k constraints, pick the c -smallest set of k items from y whose deletion causes the constraint to again become satisfied; and we delete the union of these sets from y , obtaining z . Each set has c -cost at most $\frac{k}{\gamma}c(y)$ since for each constraint i , any k elements from g^i form an eligible set for deletion, and $y \subset g^i$ consists of at least γ items. Thus $c(z) \geq c(y) - k\frac{k}{\gamma}c(y) = (1 - k^2/\gamma)c(y)$. Taking $\gamma = k^2/\epsilon$ (compared to the previous k/ϵ), we get the desired result.

5 Discussion

We believe that the main result is a nice theoretical illustration of techniques (filtering, rounding, disjunctive programming). However, it remains to be seen if it could be given useful applications. The disjunctive programming trick is definitely senseless sometimes: if we want to write an LP-based computer program to $(1 + \epsilon)$ -approximately solve multidimensional knapsack instances, it is more efficient to consider the LP corresponding to each guess separately (as in [13]) rather than solve the gigantic LP obtained by merging them together. Sometimes an LP-relative [19] (or Lagrangian-preserving [11, 1]) approximation algorithm can be used as a subroutine in ways that a non-LP-relative one could not. However, at least in [11, 19, 1], the analysis relied on LP-relative or Lagrangian-preserving analysis of the naïve LP, and an arbitrary LP would not have fared as well, and the LP we build here seems not to be useful in this way.

Finding a compact formulation for k -dimensional covering knapsack with small integrality gap and such that the LP does *not* depend on the objective function is an interesting open problem. For example, we are not aware of any polynomial-sized extended LP for 1-dimensional covering knapsack with constant integrality gap, in sharp contrast to the packing case. A partial result for k -dimensional covering knapsack is the knapsack-cover LP [6] (see also [25, 8, 7] for applications);

it has integrality gap at most $2k$, and while it is not polynomial size, it can be $(1 + \epsilon)$ -approximately separated [6] and hence $(1 + \epsilon)$ -approximately optimized [14, 16] in polynomial time.

From a theoretical perspective, it also seems challenging to find an LP for 2-dimensional (packing) knapsack where the size of the LP is a function of $1/\epsilon$ times a polynomial in n , as was done in [4] for the 1-dimensional version.

Acknowledgments

We thank Laura Sanità for helpful discussions on this topic.

References

- [1] A. Archer, M. Bateni, M. T. Hajiaghayi, and H. J. Karloff. Improved approximation algorithms for prize-collecting Steiner tree and TSP. In *Proc. 50th FOCS*, pages 427–436, 2009.
- [2] E. Balas. Disjunctive programming. In *Discrete Optimization II (Proc. Advanced Research Institute on Discrete Optimization and Systems Applications, Banff, Alta., 1977)*, volume 5 of *Annals of Discrete Mathematics*, pages 3–51. North-Holland, 1979.
- [3] D. Bienstock. Approximate formulations for 0-1 knapsack sets. *Oper. Res. Lett.*, 36(3):317–320, 2008.
- [4] D. Bienstock and B. McClosky. Tightening simple mixed-integer sets with guaranteed bounds. Manuscript, July 2008.
- [5] T. Carnes and D. B. Shmoys. Primal-dual schema for capacitated covering problems. In *Proc. 13th IPCO*, pages 288–302, 2008.
- [6] R. D. Carr, L. Fleischer, V. J. Leung, and C. A. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *Proc. 11th SODA*, pages 106–115, 2000.
- [7] D. Chakrabarty, C. Chekuri, S. Khanna, and N. Korula. Approximability of capacitated network design. arXiv:1009.5734, to appear at 15th IPCO, 2011.
- [8] D. Chakrabarty, E. Grant, and J. Knemann. On column-restricted and priority covering integer programs. In F. Eisenbrand and F. Shepherd, editors, *Integer Programming and Combinatorial Optimization*, volume 6080 of *Lecture Notes in Computer Science*, pages 355–368. Springer Berlin / Heidelberg, 2010.
- [9] A. Chandra, D. Hirschberg, and C. Wong. Approximate algorithms for some generalized knapsack problems. *Theoretical Computer Science*, 3(3):293–304, 1976.
- [10] C. Chekuri and J. Vondrák. Randomized pipage rounding for matroid polytopes. arXiv:0909.4348, September 2009.
- [11] F. A. Chudak, T. Roughgarden, and D. P. Williamson. Approximate k -MSTs and k -Steiner trees via the primal-dual method and Lagrangean relaxation. *Mathematical Programming*, 100:411–421, 2004. Preliminary version appeared in *Proc. 8th IPCO*, pages 60–70, 2001.

- [12] U. Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45, 1998. Preliminary version appeared in *Proc. 28th STOC*, pages 314–318, 1996.
- [13] A. M. Frieze and M. R. B. Clarke. Approximation algorithms for the m -dimensional 0-1 knapsack problem: Worst-case and probabilistic analyses. *European Journal of Operational Research*, 15(1):100–109, January 1984.
- [14] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *SIAM J. Comput.*, 37:630–652, May 2007. Preliminary version appeared in *Proc. 39th FOCS*, pages 300–309, 1998.
- [15] G. V. Gens and E. V. Levner. Complexity and approximation algorithms for combinatorial problems: A survey. Technical report, Central Economic and Mathematical Institute, Academy of Sciences of the USSR, Moscow, 1979.
- [16] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, 1988.
- [17] A. R. Karlin, C. Mathieu, and C. T. Nguyen. Integrality gaps of linear and semi-definite programming relaxations for knapsack. arXiv:1007.1283, to appear at 15th IPCO, 2011.
- [18] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2004.
- [19] J. Könemann, O. Parekh, and D. Pritchard. Multicommodity flow in trees: Packing via covering and iterated relaxation. Manuscript. Preliminary version appeared as *Max-Weight Integral Multicommodity Flow in Spiders and High-Capacity Trees* in *Proc. 6th WAOA*, pages 1–14, 2008.
- [20] B. Korte and R. Schrader. On the existence of fast approximation schemes. In *Proc. 4th Symp., Madison, Wisc.*, volume 4 of *Nonlinear Programming*, pages 415–437. Academic Press, 1980.
- [21] J. Lee, V. S. Mirrokni, V. Nagarajan, and M. Sviridenko. Non-monotone submodular maximization under matroid and knapsack constraints. In *Proc. 41st STOC*, pages 323–332, 2009.
- [22] H. Lenstra. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8:538–548, 1983.
- [23] G. S. Lueker. Two NP-complete problems in nonnegative integer programming. Technical Report 178, Computer Science Laboratory, Princeton University, 1975.
- [24] M. J. Magazine and M.-S. Chern. A note on approximation schemes for multidimensional knapsack problems. *Math. of Oper. Research*, 9(2):244–247, 1984.
- [25] D. Pritchard and D. Chakrabarty. Approximability of sparse integer programs. *Algorithmica*, 2010. In press. Preliminary versions at arXiv:0904.0859 and in *Proc. 17th ESA*, pages 83–94, 2009.